AD-760 437

CONTUR-A FORTRAN IV SUBROUTINE FOR THE PLOTTING OF CONTOUR LINES

George W. Hartwig, Jr.

Ballistic Research Laboratories

Prepared for:

Army Material Command

March 1973

DISTRIBUTED BY:

NTIS
National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

AD 763 0437

BRL MR 2282

BRL

AD

MEMORANDUM REPORT NO. 2282

CONTUR-A FORTRAN IV SUBROUTINE FOR THE PLOTTING OF CONTOUR LINES.

by

George W. Hartwig, Jr.

March 1973

Approved for public release; distribution unlimited.

USA BALLISTIC RESEARCH LABORATORIES
ABERDEEN PROVING GROUND, MARYLAND

35

Destroy this report when it is no longer needed.
Do not return it to the originator.

Secondary distribution of this report by originating or sponsoring activity is prohibited.

Additional copies of this report may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY *(Corporate author)* Ballistic Research Laboratories Applied Mathematics Laboratory ATTN: AMXBR-AM		2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE "CONTUR - A FORTRAN IV Subroutine for the Plotting of Contour Lines."		
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*		
5. AUTHOR(S) *(First name, middle initial, last name)* George W. Hartwig, Jr.		
6. REPORT DATE March 1973	7a. TOTAL NO. OF PAGES ~~35~~ 35	7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. b. PROJECT NO. 1T161102A14B c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) BRL MEMORANDUM REPORT NO. 2282 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY U.S. Army Materiel Command 5001 Eisenhower Avenue Alexandria, Virginia 22304	

13. ABSTRACT

In performing engineering or scientific data analysis or computations it frequently becomes necessary to examine data which is a single valued function of two independent variables. One convenient method of displaying this type of data is with contour plots. This report describes an efficient algorithm for construction of contour lines and the implementation of this algorithm as a FORTRAN IV subroutine, CONTUR.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
FORTRAN IV Subroutine Contour Lines Calcomp Plots Computer Graphics						

ib

B A L L I S T I C   R E S E A R C H   L A B O R A T O R I E S

MEMORANDUM REPORT NO. 2282

March 1973

CONTUR-A FORTRAN IV SUBROUTINE FOR THE PLOTTING OF CONTOUR LINES

George W. Hartwig, Jr.

Applied Mathematics Laboratory

RDT&E PROJECT NO. 1T161102A14B

A B E R D E E N   P R O V I N G   G R O U N D,   M A R Y L A N D

BALLISTIC RESEARCH LABORATORIES

MEMORANDUM REPORT NO. 2282

GWHartwig/cas
Aberdeen Proving Ground, Md.
March 1973

CONTUR-A FORTRAN IV SUBROUTINE FOR THE PLOTTING OF CONTOUR LINES

ABSTRACT

In performing engineering or scientific data analysis or computations it frequently becomes necessary to examine data which is a single valued function of two independent variables. One convenient method of displaying this type of data is with contour plots. This report describes an efficient algorithm for construction of contour lines and the implementation of this algorithm as a FORTRAN IV subroutine, CONTUR.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

In performing engineering or scientific data analysis it frequently becomes necessary to examine data which is a single valued function of two independent variables. A common and useful technique for displaying such data is through the use of contour plots. When three independent variables are involved any method of graphic display is cumbersome, but the plotting of contours in two dimensions for several values of the third independent variable may be the most practical alternative.

Organizations heavily involved in scientific computation utilizing digital computers frequently need to reduce results into an easily comprehendable form such as contour plots. Accordingly, it is highly desirable that an easy to use subroutine for contour plotting be available for the organization's computer users. CONTUR is such a subroutine, written in FORTRAN IV, and hence, is compatible with many digital computers in use today. The subroutine described herein was designed to work in conjunction with the California Computer Products, model 780 digital, incremental plotting system and the associated plotting subroutines in use at BRL. However, with simple modifications, CONTUR may be used with other forms of graphic display equipment.

II. ALGORITHM

The data for which contours are to be drawn is assumed to be a discrete tabulation of the single valued function

$$Z = f(x,y) \tag{1}$$

for x,y, in the range over which contours are desired. For a fixed Z, $Z=Z_0$, Eq. (1) may be written

$$Y = g(x,z_0). \tag{2}$$

Preceding page blank

In this form the curve is called a contour and in general a different contour would occur for each value of Z_o. Usually the function, $f(x,y)$ is not known, the data arising either from experiment or by numerical approximation techniques. Hence, the explicit expression as a function of x and Z_o, Eq. (2), is not available and a numerical procedure for determining the contours is necessary.

The algorithm described below represents a significant simplification of the algorithm described by James Downing [1].

The algorithm is derived by focusing attention on four adjacent data points $Z_{i,j}$, $Z_{i+1,j}$, $Z_{i,j+1}$ and $Z_{i+1,j+1}$ where the coresponding independent variables have the values (X_i, Y_j), (X_{i+1}, Y_i) etc. Assuming the data contains I points in the x direction and J points in the Y direction, the algorithm must be applied to $N = (I-1)(J-1)$ cells.

Within such a cell, Figure 1, the center point is located and assigned a Z value equal to the average of the four Z values at the corners. These five points are then connected with line segments which are in turn numbered one through eight in a clockwise direction.

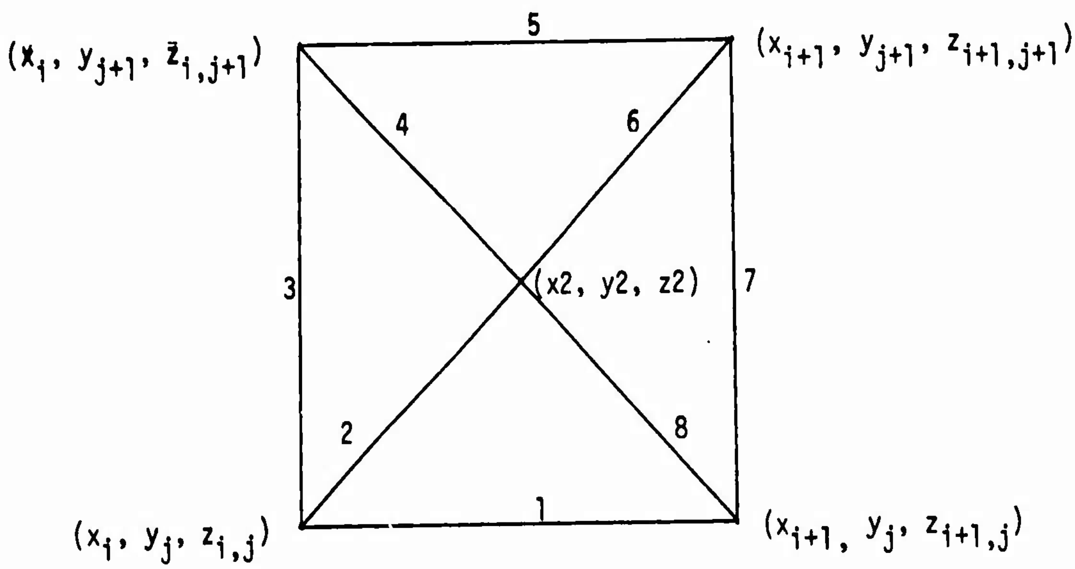


Figure 1. A Typical Cell.

Each segment is then tested to see if the required contour intersects with it, in the following manner. Starting with segment one, the contour value Z_o is subtracted from the end points.

$$T_1 = Z_{i,j} - Z_o$$
$$T_2 = Z_{i+1,j} - Z_o \tag{3}$$

If the quantity

$$\Delta = T_1 \cdot T_2 \tag{4}$$

is greater than zero, the entire segment is either above or below Z_o, if Δ equals zero, either $Z_{i,j}$ or $Z_{i+1,j}$ is equal to Z_o, and if Δ is less than zero the contour intersects the segment. In this last case the point of intersection, x_o is found by linear interpolation (see Figure 2) with x_o given by

$$x_o = (Z_o - Z_{i,j})(x_{i+1} - x_i)/(Z_{i+1,j} - Z_{i,j}) + x_i. \tag{5}$$

The x and y values of this intersection are then stored in temporary arrays PX and PY.

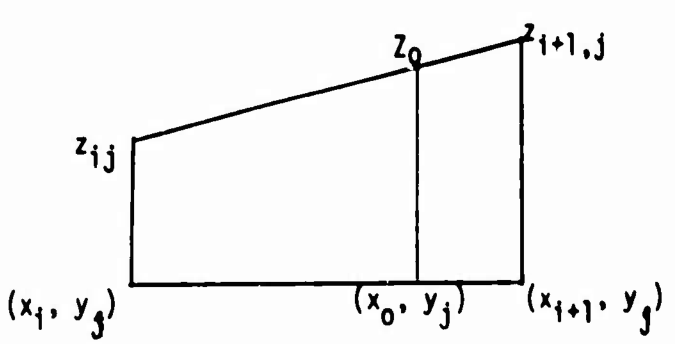


Figure 2. The Interpolation Scheme.

The procedure is then repeated for segments two through eight. When segment eight is completed, the points stored in PX and PY are plotted and the next set of points are considered.

Before the ordered pairs (PX, PY) can be plotted successfully there are several conditions which must be tested for and if present, properly handled. (1) If all four of the cell's corner points are equal to Z_o, no points should be plotted. (2) When the contour intersects segment eight, the PX and PY arrays must be reordered. The reason for this becomes obvious when one remembers that the segments are tested in a clockwise direction. For instance, assume CONTUR finds intersections on segments one, seven and eight. Plotting these points as originally stored would result in an extraneous line being drawn. See Figure 3. By simply rearranging the points so that they are stored seven, eight, one, the correct contour is drawn.

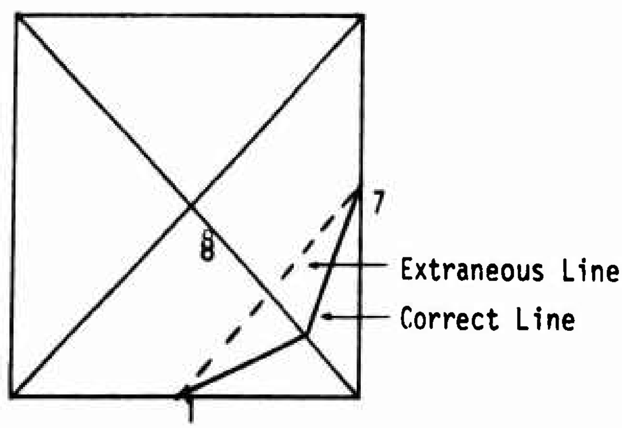


Figure 3. Error Condition 2.

(3) Provision is also made for the case where two contours of the same value pass through the cell. This occurs only when two opposite Z values are greater than Z_o and the other two points are less than Z_o. By noting if the center point, Z2, is greater than or less than Z_o, the paths taken by the contours are specifically known and are plotted as a special case. See Figure 4.

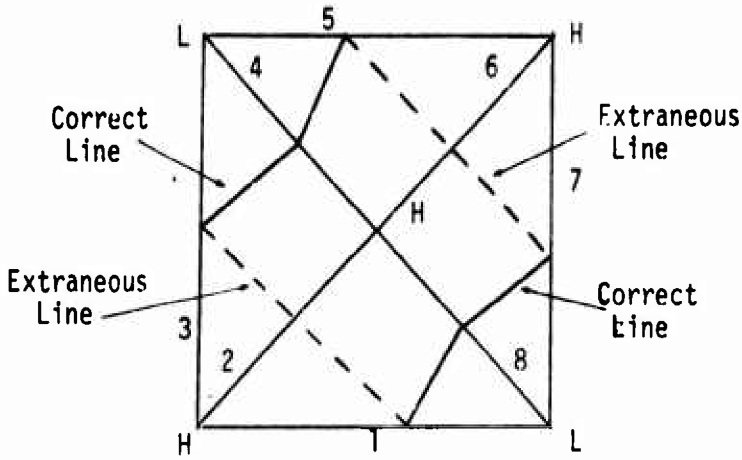


Figure 4. Error Condition 3.

III. THE SUBROUTINE

CONTUR is accessed through the statement

CALL CONTUR (Z,X,Y,IS,IY,DZ,NZ,IZ).

Z,X,Y are the arrays containing the values Z_{ij}, X_i and Y_j, respectively. IX and IY are the number of points in the X dimension and Y dimension. The subroutine requires that Z be of dimensions (IZ,IY). DZ(NZ) is a one dimensional array containing the Z values at which contours are desired. NZ is the number of these values. The declared number of rows in the Z array is IZ.

Since the subroutine uses just four data points at a time, requiring no knowledge of where it has been or where it is going, enormous amounts of data can be handled by reloading the Z array and calling CONTUR several times with different portions of the data.

The computer time required by CONTUR depends on the size of the Z array, the number of values at which contours are desired and the smoothness or irregularity of the data, with time increasing for large arrays, large numbers of contour values and irregular data. For some typical times see the examples of contour output.

In order to keep the subroutine as efficient and machine independent as possible, no labeling of contours is done, nor are any borders or titles plotted in CONTUR. The user must initialize the plot routines and set scales prior to calling CONTUR. PLTCCD is a predefined subroutine on the Ballistic Research Laboratories BRLESC computers that generates input data for the Cal Comp 780 digital, incremental plotting system, and must be replaced for use of CONTUR on other computer systems.

BRLESC users should note that the positioning on the plotter page and scales used by CONTUR are determined by the latest reference to PLTCCS. [1] Thus, it may be necessary to reset the plotting scales before calling CONTUR.

IV. CONTUR INPUT VARIABLES

Z(IZ,IY)	-	is a two dimensional array containing the functional values of the data.
X(IX)	-	is a one dimensional array containing the values of one of the independent variables.
Y(IY)	-	is a one dimensional array containing the values of the other independent variable.
IX	-	is the number of elements in the X array.
IY	-	is the number of elements in the Y array.
DZ(NZ)	-	is a one dimensional array containing the Z values at which contours are desired.
NZ	-	is the number of elements in the DZ array.
IZ	-	the number of declared rows in the Z array.

V. PLTCCD INPUT VARIABLES

N=1,L=0	-	signifies that a line plot is to be drawn.
PX(I),PY(J)	-	are arrays containing the X and Y coordinates to be plotted. The First point plotted is Px(I),PY(J).
K	-	is the number of data pairs to be plotted
M=0	-	causes the subroutine to start a new curve with the point (PX(I), PY(J)).
M=1	-	causes the subroutine to continue the curve plotted by the previous PLTCCD entry.

REFERENCES

1. Downing, James A., "The Automatic Construction of Contour Plots with Applications to Numerical Analysis Research," The University of Texas Computation Center, Austin, Texas, January 1966.

2. Coleman, Monte W., Lanahan, John V., "BRLESC FORTRAN Plotting Subroutines," ARDC Technical Report No. 6. July 1970.

3. Nagy, Nicholas J. (III), "The Graphical Representation of Two Variable Data", Los Alamos Scientific Laboratory, Report No. LA-4796, November 1971.

4. Lintner, M. A., "Proj-Algorithm and Computer Programs for the Hidden Line Problem for Single Valued Surfaces", Idaho Nuclear Corporation, December 1969.

APPENDIX A

Examples of CONTUR Output

Figure A-1 is a three dimensional graph of the function

$$Z = \mathrm{SIN}\ (x + y)/(1 + (x - y)^2) \qquad \text{A-1}$$

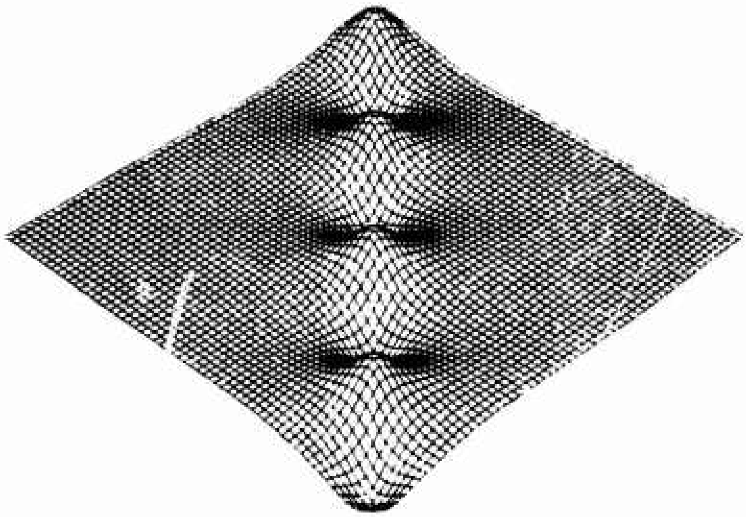
as plotted by the subroutine GRAF3D(3). Although this plot is interesting and demonstrates general trends, it is virtually impossible to retrieve any useful quantitative information from it. Figure A-2 is a contour plot as drawn by CONTUR of the same data. The contour lines are at values of .1, .4, .6, and .9. The Z array contains 3600 points and CONTUR required 14.4 seconds on the BRLESC II computer to generate the curves.

Figures A-3 and A-4 represent experimental data. Again the contour plot is the more analytically useful, although not as esthetically pleasing as the 3-D plot. The PR01 subroutine (4) was used in this case to generate the three dimensional plot.

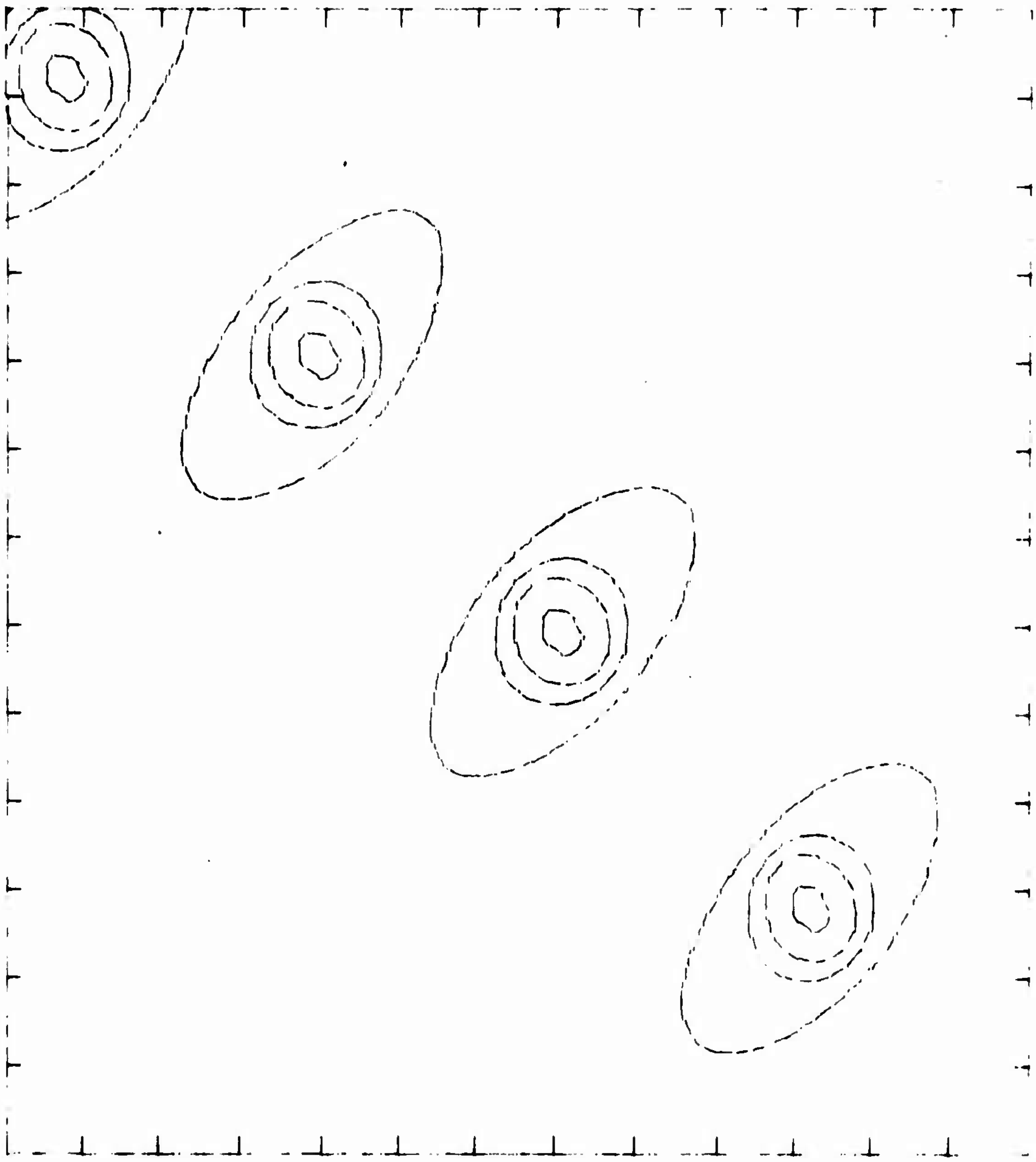
Figures A-5,6,7 are included to demonstrate the results of allowing the data gride to become too coarse. Figure A-5 is the 3-D representation of

$$Z = \left|\mathrm{SIN}\ (\sqrt{x^2 + y^2})/(\sqrt{x^2 + y^2}\right| \qquad -20 \leq x,y \leq 20.$$

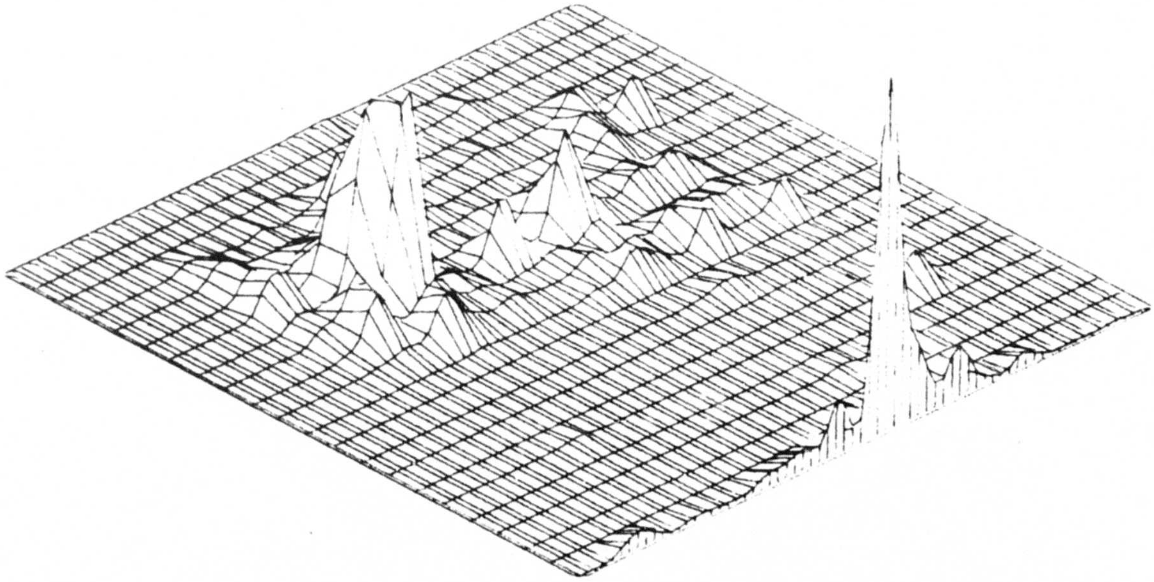
Both Figures A-6 and A-7 are contour plots of the above function with contour lines drawn at Z values of .1, .4, .6, and .9. In Figure A-6 the grid contains 10000 points and the representation is accurate. The grid in Figure A-7 contains only 2500 points and the interpolation scheme is no longer sufficiently accurate to portray the function correctly. The BRLESC I computer required 60.6 seconds to generate A-6 and 19.2 seconds for A-7.



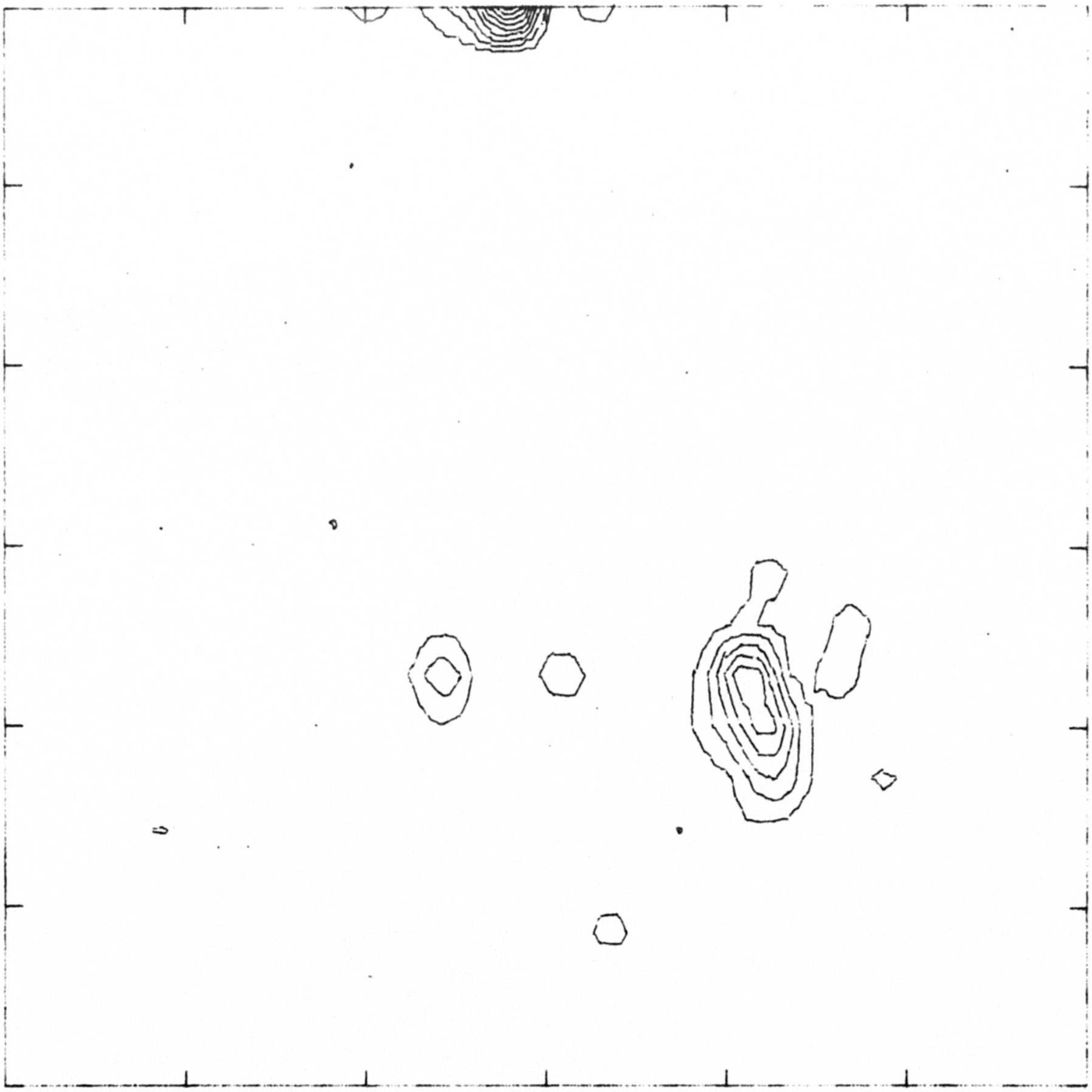
A-1. 3-D Plot of $Z = SIN(x + y)/(1 + (x - y)^2)$



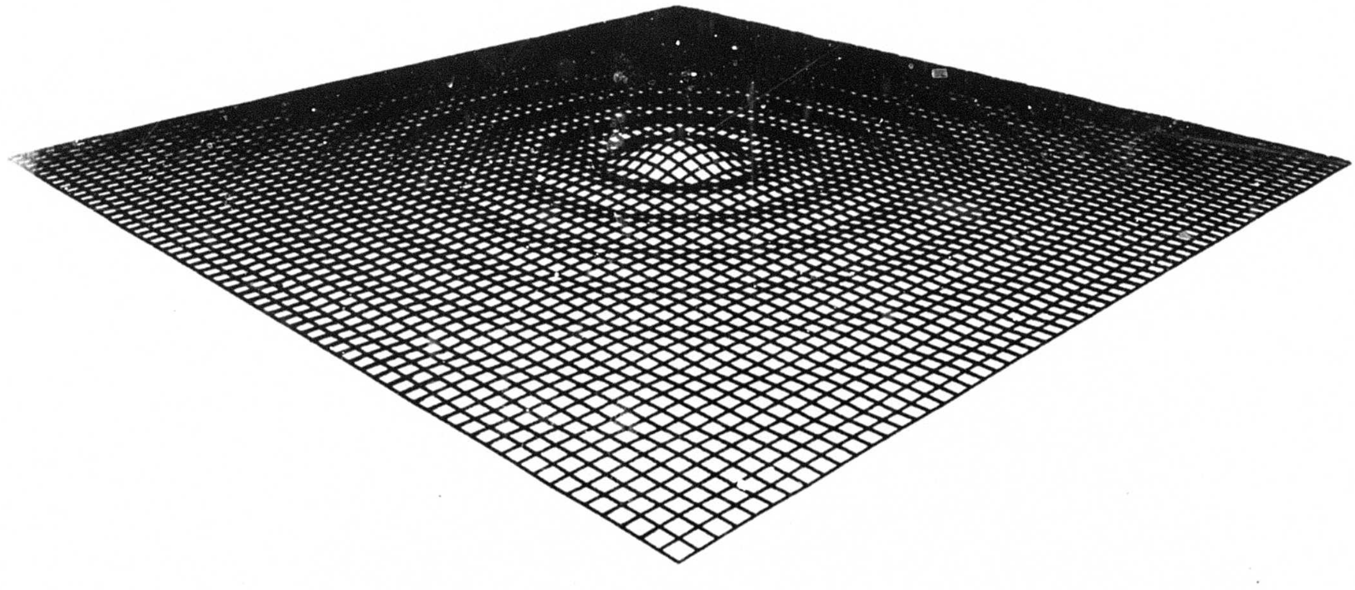
A-2. Contours of Z = SIN (x [illegible])/(1 + (x - Y)^2)



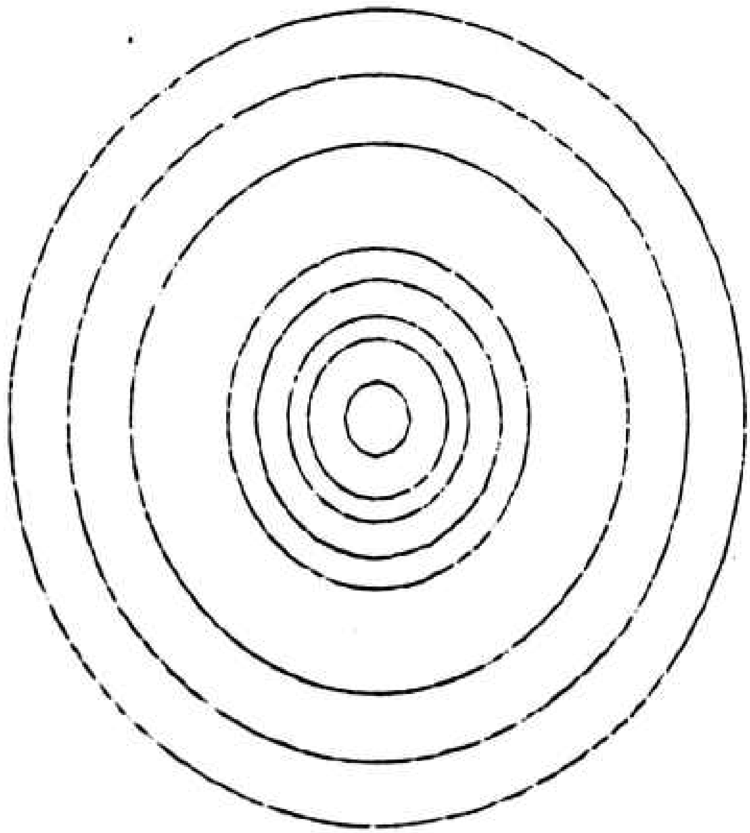
A-3. 3-D Plot of Experimental Data



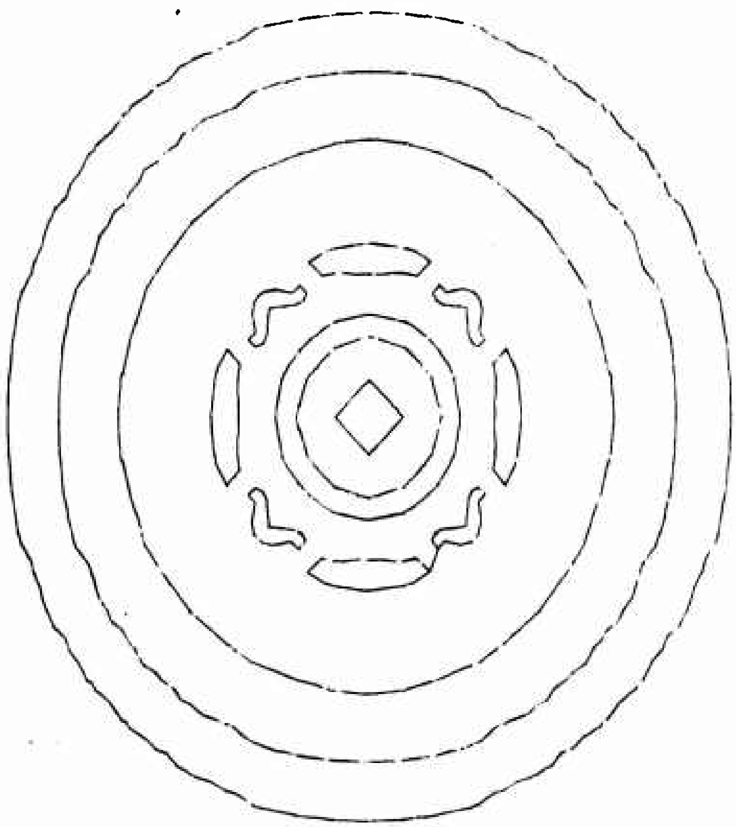
A-4. Contours of Experimental Data



A-5. 3-D Plot of $Z = |\mathrm{SIN}\ (\sqrt{x^2 + y^2})/\sqrt{x^2 + y^2}\ |$



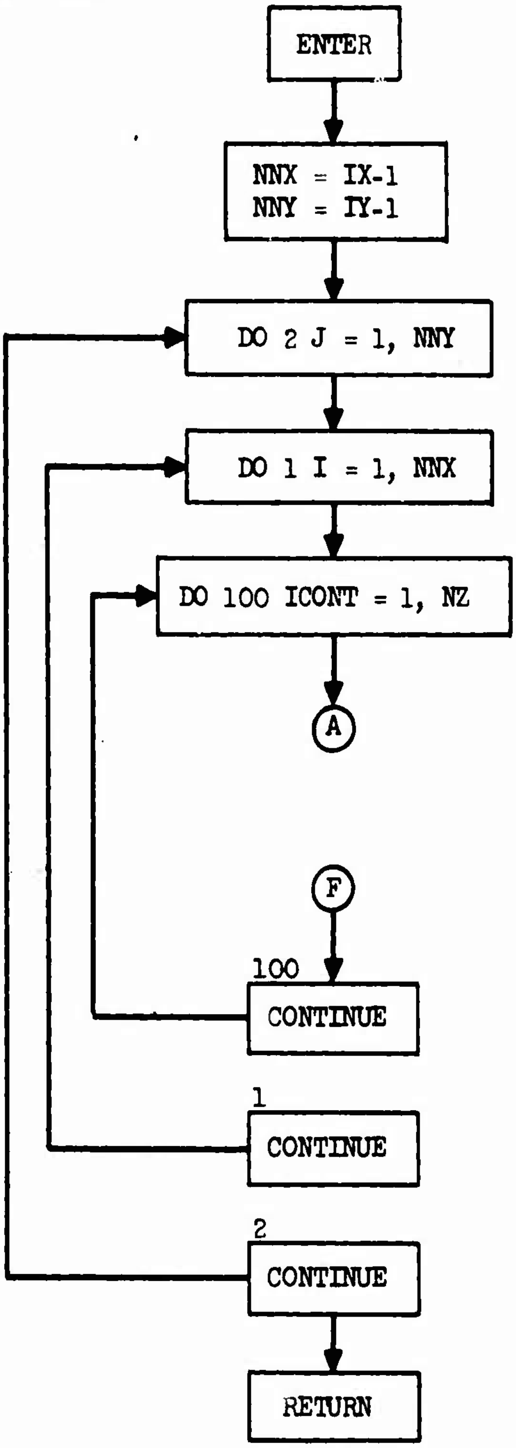
A-6. Dense Grid Contours of $Z = |\mathrm{SIN}\ (\sqrt{x^2 + y^2})/\sqrt{x^2 + y^2}\,|$

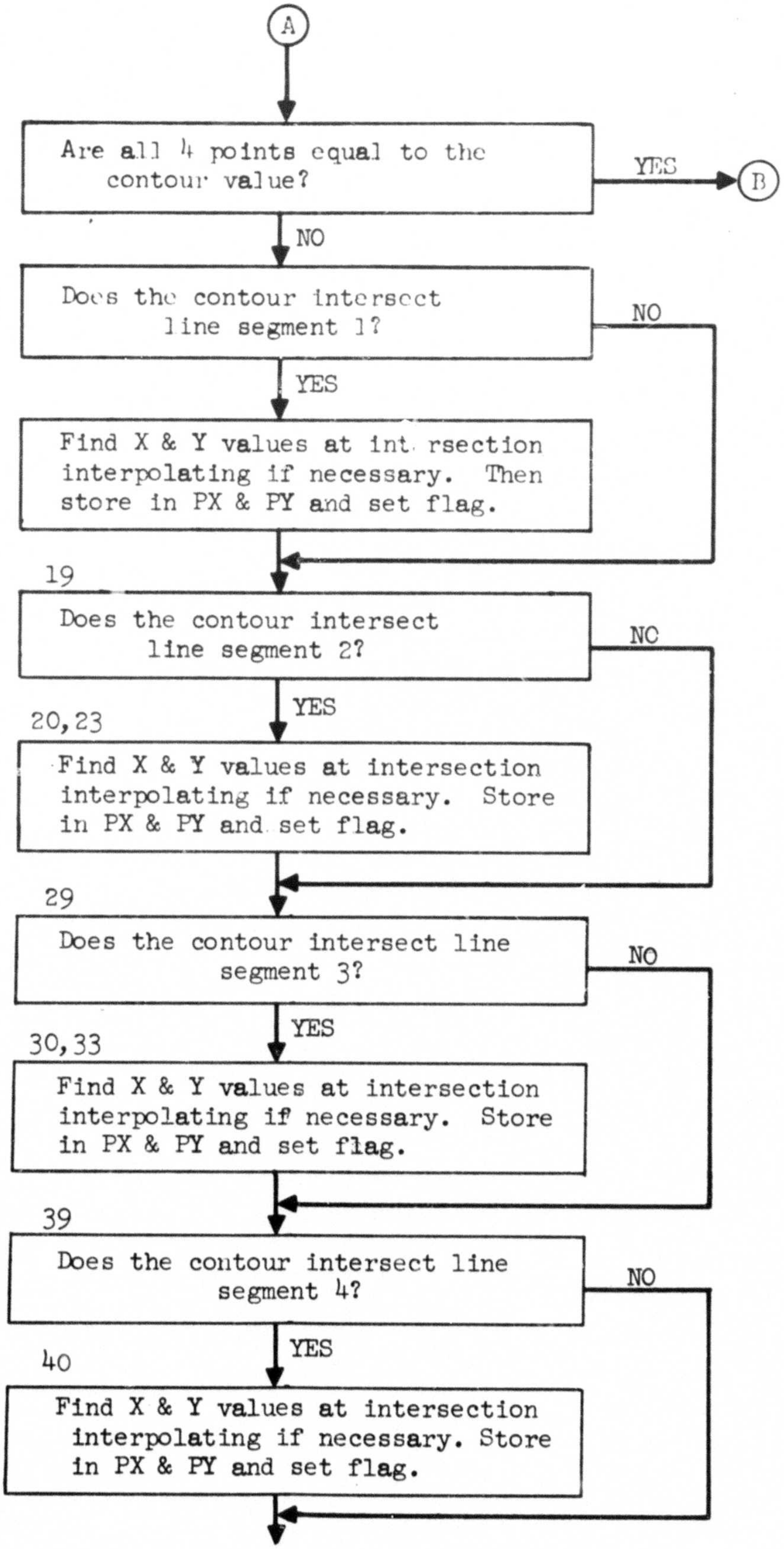


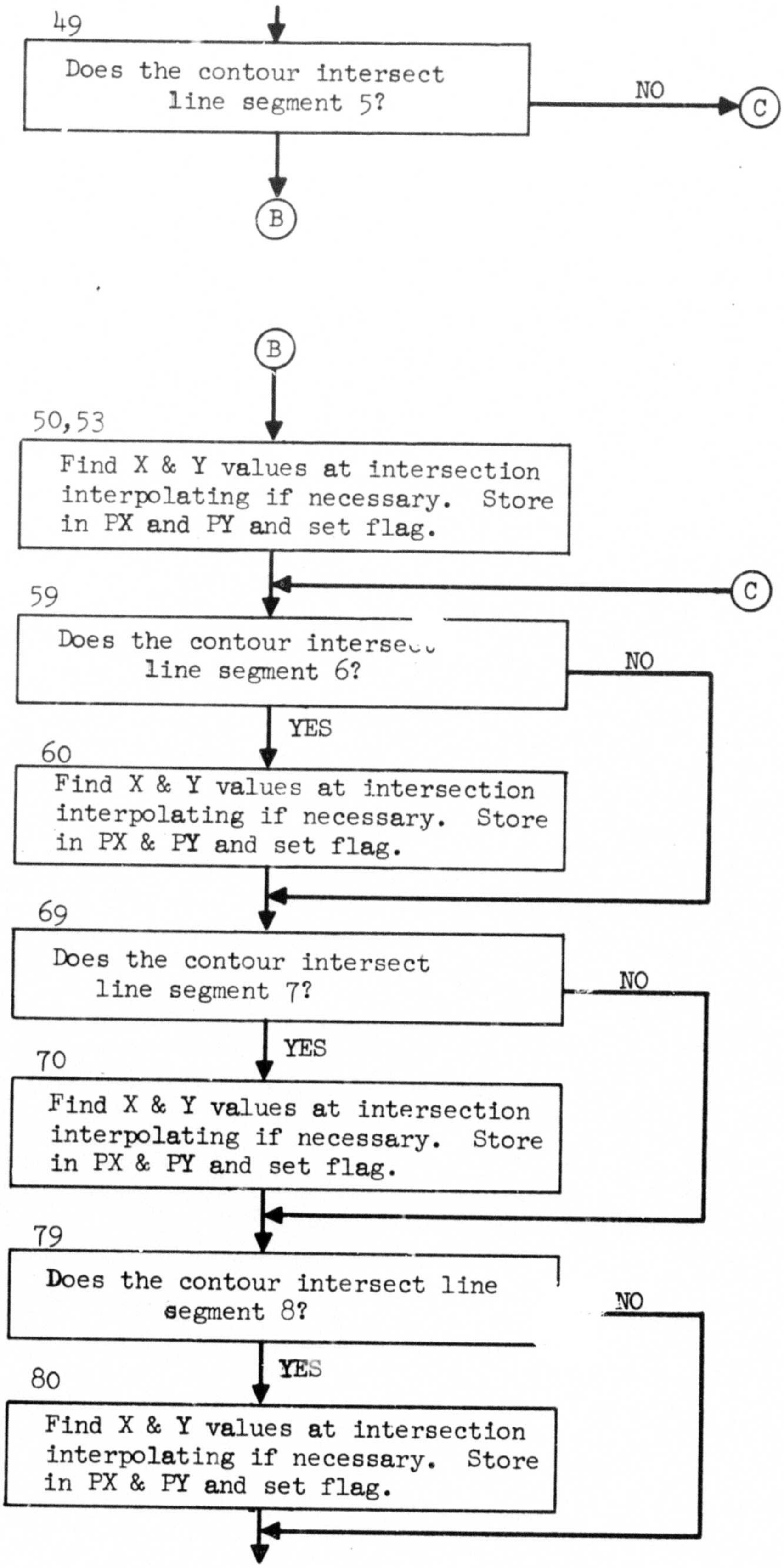
A-7. Coarse Grid Contours of $Z = \left| \sin(\sqrt{x^2 + y^2}) / \sqrt{x^2 + y^2} \right|$

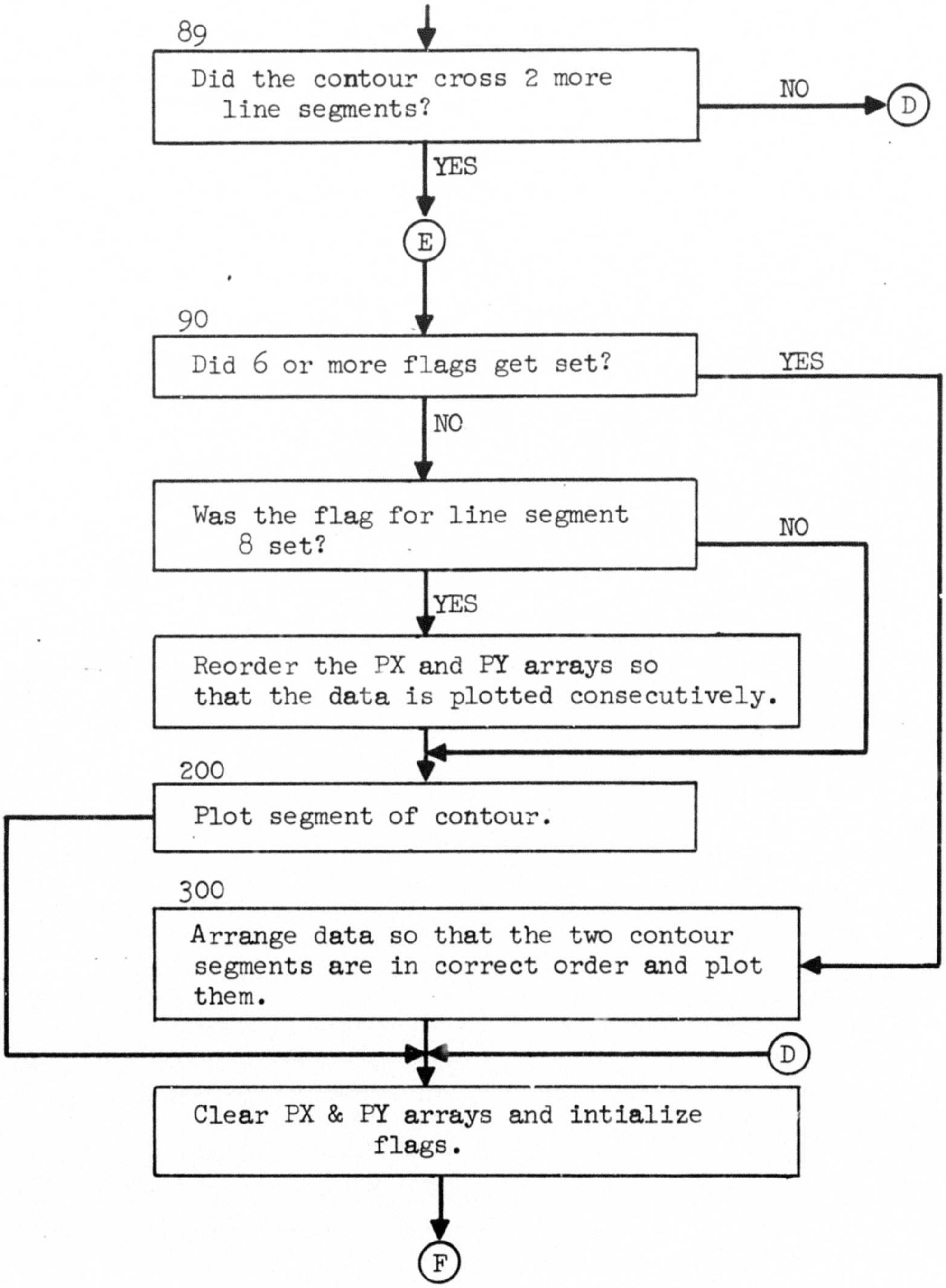
APPENDIX B

Flow Chart of Contur




A
Are all 4 points equal to the contour value?
YES
B
NO
Does the contour intersect line segment 1?
NO
YES
Find X & Y values at intersection interpolating if necessary. Then store in PX & PY and set flag.
19
Does the contour intersect line segment 2?
NO
YES
20,23
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.
29
Does the contour intersect line segment 3?
NO
YES
30,33
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.
39
Does the contour intersect line segment 4?
NO
YES
40
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.


49
Does the contour intersect line segment 5?
NO
C
B
B
50,53
Find X & Y values at intersection interpolating if necessary. Store in PX and PY and set flag.
C
59
Does the contour intersect line segment 6?
NO
YES
60
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.
69
Does the contour intersect line segment 7?
NO
YES
70
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.
79
Does the contour intersect line segment 8?
NO
YES
80
Find X & Y values at intersection interpolating if necessary. Store in PX & PY and set flag.


89
Did the contour cross 2 more line segments?
NO
D
YES
E
90
Did 6 or more flags get set?
YES
NO
Was the flag for line segment 8 set?
NO
YES
Reorder the PX and PY arrays so that the data is plotted consecutively.
200
Plot segment of contour.
300
Arrange data so that the two contour segments are in correct order and plot them.
D
Clear PX & PY arrays and intialize flags.
F

```
      SUBROUTINE CONTUR (Z, X, Y, IX, IY, DZ, NZ, IZ)
C
      DIMENSION X(IX),Y(IY),Z(IZ,IY),PX(8),PY(8),KCHK(8),DZ(NZ)
C
C  THIS SUBROUTINE PLOTS NZ CONTOURS AT DZ VALUES.
C
C  X AND Y ARE ONE DIMENSIONAL ARRAYS OF LENGTH IX AND IY, RESPECTIVELY.
C
C  Z IS A TWO DIMENSIONAL ARRAY OF SIZE (IX,IY).
C
C  DZ IS A ONE DIMENSIONAL ARRAY OF LENGTH NZ IN WHICH THE Z VALUES AT
C    WHICH CONTOURS ARE DESIRED ARE PLACED.
C
C  THIS VERSION OF CONTUR WAS COMPLETED IN FEBRUARY 1973.
C
      IC=0
      NNX=IX-1
      NNY=IY-1
      DO 2 J=1,NNY
      DO 1 I=1,NNX
      DO 100 ICONT=1,NZ
      Z0=DZ(ICONT)
C
C   IF ALL FOUR DATA POINTS ARE EQUAL TO Z0, DO NOT PLOT ANY LINES FOR
C     THIS CELL.
C
      IF(Z(I,J).EQ.Z0.AND.Z(I+1,J).EQ.Z0.AND.Z(I,J+1).EQ.Z0.AND.
     1Z(I+1,J+1).EQ.Z0) GOTO 100
C
C  TEST SEGMENT 1 FOR AN INTERSECTION WITH THE CONTOUR LINE.
C
      T1=Z(I,J)-Z0
      T2=Z(I+1,J)-Z0
      D=T1*T2
      IF(D)10,11,19
   10 IC=IC+1
      PX(IC)=-T1*(X(I+1)-X(I))/(Z(I+1,J)-Z(I,J))+X(I)
      PY(IC)=Y(J)
      KCHK(1)=1
      GOTO 19
   11 IF(T1.NE.0.0) GOTO 13
      IC=IC+1
      PX(IC)=X(I)
      PY(IC)=Y(J)
      KCHK(1)=1
      IF(T2)19,13,19
   13 IC=IC+1
      PX(IC)=X(I+1)
      PY(IC)=Y(J)
      KCHK(1)=1
C
C  TEST SEGMENT 2 FOR AN INTERSECTION WITH THE CONTOUR LINE.
C
   19 T3=.25*(Z(I,J)+Z(I+1,J)+Z(I,J+1)+Z(I+1,J+1))-Z0
      D=T1*T3
      X2=(X(I+1)+X(I))*.5
      Y2=(Y(J+1)+Y(J))*.5
      Z2=T3+Z0
      IF(D)20,23,29
   20 IC=IC+1
```

```
      PX(IC)=-T1*(X2-X(I))/(Z2-Z(I,J))+X(I)
      PY(IC)=-T1*(Y2-Y(J))/(Z2-Z(I,J))+Y(J)
      KCHK(2)=1
      GOTO 29
   23 IF(T3.NE.0.0) GOTO 29
      IC=IC+1
      PX(IC)=X2
      PY(IC)=Y2
      KCHK(2)=1
C
C  TEST SEGMENT 3 FOR AN INTERSECTION WITH THE CONTOUR LINE.
C
   29 T2=Z(I,J+1)-Z0
      D=T1*T2
      IF(D)30,33,39
   30 IC=IC+1
      PX(IC)=X(I)
      PY(IC)=-T1*(Y(J+1)-Y(J))/(Z(I,J+1)-Z(I,J))+Y(J)
      KCHK(3)=1
      GOTO 39
   33 IF(T2.NE.0.0) GOTO 39
      IC=IC+1
      PX(IC)=X(I)
      PY(IC)=Y(J+1)
      KCHK(3)=1
C
C  TEST SEGMENT 4 FOR AN INTERSECTION WITH THE CONTOUR LINE.
C
   39 T1=Z(I,J+1)-Z0
      D=T1*T3
      IF(D)40,49,49
   40 IC=IC+1
      PX(IC)=-T1*(X2-X(I))/(Z2-Z(I,J+1))+X(I)
      PY(IC)=-T1*(Y2-Y(J+1))/(Z2-Z(I,J+1))+Y(J+1)
      KCHK(4)=1
C
C  TEST SEGMENT 5
C
   49 T2=Z(I+1,J+1)-Z0
      D=T1*T2
      IF(D)50,53,59
   50 IC=IC+1
      PX(IC)=-T1*(X(I+1)-X(I))/(Z(I+1,J+1)-Z(I,J+1))+X(I)
      PY(IC)=Y(J+1)
      KCHK(5)=1
      GOTO 59
   53 IF(T2.NE.0.0) GOTO 59
      IC=IC+1
      PX(IC)=X(I+1)
      PY(IC)=Y(J+1)
      KCHK(5)=1
C
C  TEST SEGMENT 6
C
   59 D=T2*T3
      IF(D)60,69,69
   60 IC=IC+1
      PX(IC)=-T2*(X2-X(I+1))/(Z2-Z(I+1,J+1))+X(I+1)
      PY(IC)=-T2*(Y2-Y(J+1))/(Z2-Z(I+1,J+1))+Y(J+1)
      KCHK(6)=1
```

```
C
C  TEST SEGMENT 7
C
   69 T1=T2
      T2=Z(I+1,J)-Z0
      D=T1*T2
      IF(D)70,79,79
   70 IC=IC+1
      PX(IC)=X(I+1)
      PY(IC)=-T1*(Y(J)-Y(J+1))/(Z(I+1,J)-Z(I+1,J+1))+Y(J+1)
      KCHK(7)=1
C
C  TEST SEGMENT 8
C
   79 D=T2*T3
      IF(D)80,89,89
   80 IC=IC+1
      PX(IC)=-T3*(X(I+1)-X2)/(Z(I+1,J)-Z2)+X2
      PY(IC)=-T3*(Y(J)-Y2)/(Z(I+1,J)-Z2)+Y2
      KCHK(8)=1
   89 IF(IC.GE.2) GOTO 90
      GOTO 201
C
C  THIS SECTION OF CODING ORDERS THE DATA SO THAT NO OVERLAPPING OR
C     BACKTRACKING OCCURS DURING PLOTTING.
C
   90 IF(IC.GE.6) GOTO 300
      IF(KCHK(8).NE.1) GOTO 200
      DO 101 L=1,7
      IF(KCHK(L).NE.1) GOTO 101
      IC=IC+1
      PX(IC)=PX(1)
      PY(IC)=PY(1)
      IC=IC-1
      DO 102 M=1,IC
      PX(M)=PX(M+1)
  102 PY(M)=PY(M+1)
      IF((MOD(L,2).EQ.1).AND.KCHK(L).EQ.1) GOTO 200
  101 CONTINUE
C
C  PLOT DATA FOR THE USUAL CELL.
C
  200 CALL PLTCCD (1,0,PX(1),PY(1),IC,0)
      GOTO 201
C
C  THIS SECTION OF CODING DOES THE ORDERING AND PLOTTING IF 2 CONTOUR
C     LINES PASS THROUGH THE CELL
C
  300 IF(Z(I,J).GT.Z0.AND.Z(I+1,J+1).GT.0.0) GOTO 301
      IF(Z2.GT.Z0) GOTO 303
  302 N=IC+1
      PX(N)=PX(1)
      PY(N)=PY(1)
      CALL PLTCCD (1,0,PX(5),PY(5),3,0)
      CALL PLTCCD (1,0,PX(2),PY(2),3,0)
      GOTO 310
  303 CALL PLTCCD (1,0,PX(1),PY(1),3,0)
      CALL PLTCCD (1,0,PX(4),PY(4),3,0)
      GOTO 310
  301 IF(Z2.GT.Z0) GOTO 302
```

```
      GOTO 303
  310 CONTINUE
C
C  CLEAR WORKING ARRAYS AND INITIALIZE FLAGS
C
  201 IC=0
      DO 8 MN=1,8
      PX(MN)=0.
      PY(MN)=0.
    8 KCHK(MN)=0
  100 CONTINUE
    1 CONTINUE
    2 CONTINUE
      RETURN
      END
```